N91-21781

The Simplification of Fuzzy Control Algorithm and Hardware Implementation

Z. Q. Wu, P. Z. Wang and H. H. Teh
Institute of Systems Science
National University of Singapore
Heng Mui Keng Terrace,Kent Ridge
Singapore 0511

Abstract

The conventional inference composition algorithm of fuzzy controller is very time and memory consuming. As a result, it is difficult to do real time fuzzy inference and most fuzzy controllers are realized by look-up tables. In this paper we derived a simplified algorithm using the defuzzification mean of maximum. This algorithm takes shorter computation time and needs less memory usage, thus making it possible to compute the fuzzy inference on real time and easy to tune the control rules on line. The responsibility of this algorithm is proved mathematically in this paper.

Fuzzy controller has been highly developed and come to a new stage of hardware implementation. Many fuzzy controllers(or so called fuzzy inference machines) in hardware have been available in the market. The conventional fuzzy inference algorithm on which most fuzzy controller based on is too complicated. Further, its hardware implementation is very expensive and of a large volume, and the inference speed is limited. Reducing its cost and volume and improving its inference speed are very important to this technology. In this paper we also describe a hardware implementation based on the above simplified fuzzy inference algorithm.

1. Fuzzy controller algorithm

Assume that the fuzzy controller has two inputs and a single output as shown in Figure 1,

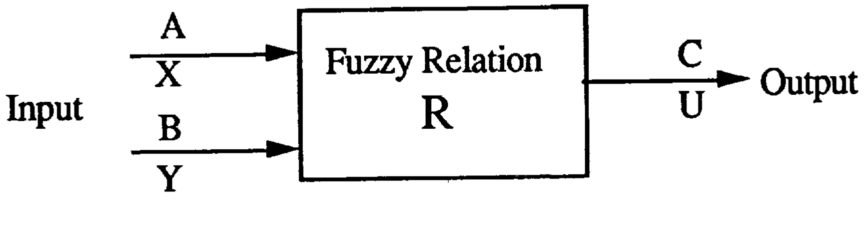

Fig.1 The block graph of fuzzy controller

where A and B are the linguistic variables of the inputs, with universe of discourse X and Y respectively, and C is the linguistic variable of the output, with universe of discourse U. We emphasize here that X and Y are not necessarily continuous on the real line R, but arbitrary subsets of R.

Let the sets of linguistic values concerning with A, B and C respectively be as follows

$$\{A_i\} \in \mathcal{F}(X),\ (i \in I) \tag{1}$$
$$\{B_j\} \in \mathcal{F}(Y),\ (j \in J) \tag{2}$$
$$\{C_k\} \in \mathcal{F}(U),\ (k \in K) \tag{3}$$

where I={1, 2, ···, m}, J={1,2, ···, n}, K={1, 2, ···, h}, and $\mathcal{F}(X)$ represents the fuzzy power set of X.

The fuzzy control rules are described in terms of a group of multi-complexed fuzzy implications as follows:

$$\begin{gathered}\text{If A is } A_i \text{ and B is } B_j \text{ then C is } C_k,\\ (i \in I, j \in J, k = \varphi(i, j) \in K)\end{gathered} \tag{4}$$

The above fuzzy implications can be translated into a three-dimensional relation R according to the fuzzy Compositional Rule of Inference(CRI method).

Definition 1.

$$\begin{gathered} R \equiv \bigcup_{i,j} (A_i \times B_j \times C_k)\\ R \in \mathcal{F}(X \times Y \times U),\\ R(x, y, u) = \bigvee_{i,j} (A_i(x) \wedge B_j(y) \wedge C_k(u)).\\ (k = \varphi(i, j) \in K) \end{gathered} \tag{5}$$

Suppose that the inputs of the fuzzy controller at a certain instance are fuzzy sets $A^* \in \mathcal{F}(X)$ and $B^* \in \mathcal{F}(Y)$, according to the CRI method, the output of the controller will be the fuzzy set denoted by $C^* \in \mathcal{F}(U)$, i.e

$$C^* = (A^* \times B^*) \circ R$$

$$C^*(u) = \sup_{\substack{x \in X \\ y \in Y}} (A^*(x) \wedge B^*(y) \wedge R(x, y, u))$$

$$= \sup_{\substack{x \in X \\ y \in Y}} ((A^*(x) \wedge B^*(y)) \wedge (\bigvee_{i,j} (A_i(x) \wedge B_j(y) \wedge C_{\varphi(i, j)}(u))))$$

$$= \sup_{\substack{x \in X \\ y \in Y}} (\bigvee_{i,j} ((A^*(x) \wedge A_i(x)) \wedge (B^*(y) \wedge B_j(y)) \wedge C_{\varphi(i, j)}(u)))$$

$$= \bigvee_{i,j} \sup_{x \in X}((A^*(x) \wedge A_i(x)) \wedge \sup_{y \in Y}(B^*(y) \wedge B_j(y)) \wedge C_{\varphi(i, j)}(u) \qquad (6)$$

In actual applications the inputs of the controller (i.e the observed values of the controlled process) are some definite real numbers. Suppose in a certain instance the observed value is a pair (x_o, y_o), then the fuzzy sets of inputs A^* and B^* are as follows,

$$A^*(x) = \begin{cases} 1, & x=x_o \\ 0, & x \neq x_o \end{cases}, \quad B^*(y) = \begin{cases} 1, & y=y_o \\ 0, & y \neq y_o \end{cases} \qquad (7)$$

so that

$$\sup_{x \in X}(A^*(x) \wedge A_i(x)) = A_i(x_o) \qquad (8)$$

$$\sup_{y \in Y}(B^*(y) \wedge B_j(y)) = B_j(y_o) \qquad (9)$$

therefore

$$C^*(u) = \bigvee_{i,j} (A_i(x_o) \wedge B_j(y_o)) \wedge C_{\varphi(i, j)}(u) \qquad (10)$$

$$(i \in I, j \in J, \varphi(i, j) \in K)$$

2. The responsibility of the fuzzy controller

The responsibility of a fuzzy controller has been defined and analyzed in depth by P. Z. Wang and S. P. Lou[3], here we discuss the responsibility of fuzzy controller under a weaker condition.

Definition 2 For a set of linguistic values concerning with A is $\{A_i\}(i \in I) \in \mathcal{F}(X)$, $I=\{1, 2, \cdots, m\}$, where A_i is a normally distributed fuzzy set, there exists m+1 real numbers

$$r_o < r_1 < r_2 < \ldots < r_m ,$$

such that for any given $x \in (r_{i-1}, r_i)$, if $j \neq i$, $A_j(x) < A_i(x)$ (see Figure 2). We called $J_i = (r_{i-1}, r_i) \subseteq X$, the interval of A_i, $i \in I$, and $N^1 = \{ r_i \}$ the net of A, where J_i and N^1 satisfy the following:

$$J_i \cap J_j = \Phi \; , (i \neq j) \tag{11}$$

$$\bigcup_{i=1}^{m} J_i = X - \{ r_i \} \tag{12}$$

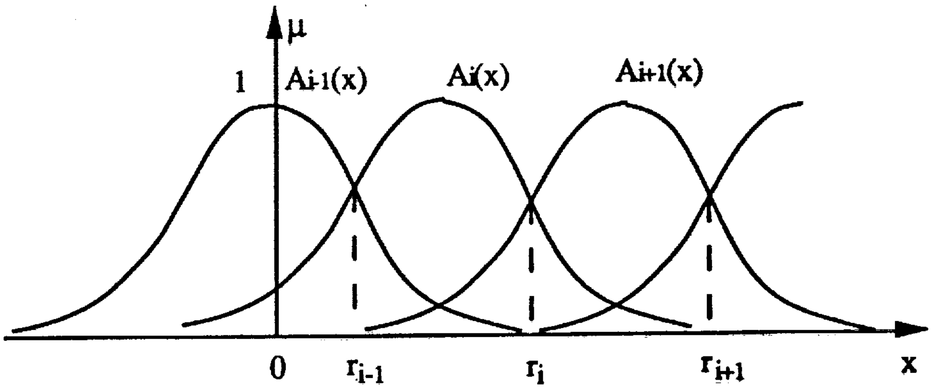

Figure 2 Membership functions of fuzzy sets A_i

For $\{A_i\} \in \mathcal{F}(X)$, $\{B_j\} \in \mathcal{F}(Y)$, we have (13)

$$(A_i \times B_j)(x, y) \equiv A_i(x) \wedge B_j(y)$$

$$\forall\, i \in I, \forall\, j \in J, I=\{1, 2, \cdots, m\}, J=\{1,2, \cdots, n\}. \tag{14}$$

If there exist nets $N^1 = \{r'_i\}$, $N^2 = \{r''_j\}$ and intervals $\{J'_i\}$, $\{J''_j\}$ for A and B respectively, then

$$(x, y) \in J_{st} \longrightarrow A_s(x) \wedge B_t(y) > A_i(x) \wedge B_j(y) \qquad (15)$$
$$((s, t) \neq (i, j))$$

where $J_{ij} \equiv J'_i \times J''_j$, is called the interval of $(A_i \times B_j)$.
In fact,
$(x, y) \in J_{st} \longrightarrow x \in J'_s, y \in J''_t \longrightarrow A_s(x) > A_i(x), B_t(y) > B_j(y)$
$(s, t) \neq (i, j)$
$\longrightarrow \min(A_s(x), B_t(y)) > \min(A_i(x), B_j(y))$
$\longrightarrow A_s(x) \wedge B_t(y) > A_i(x) \wedge B_j(y)$ (16)

we define the net of A × B as

$$N \equiv \{(x, y) \mid x = r'_i, y = r''_j\} \qquad (17)$$

According to the definition of responsibility of fuzzy controller given in [3], we state the following definition with slight changes.

Definition 3 A Fuzzy controller is said to be responsive if there exists an interval $L \subseteq (-\infty, +\infty)$ such that

$$L = \{u \mid C^*(u) = \underset{u \in U}{\text{hgt}}\, C^*(u)\}. \qquad (18)$$

where hgtC*(u) is the height of fuzzy set C*(u) and L is the responsive interval.

If a fuzzy controller is responsive, the output of the controller, according to the defuzzification mean of maximum, is

$$u_o = M(L) \quad (u_o \in U), \qquad (19)$$

where M(L) means the mid-point of L.

Theorem 1, A given fuzzy controller is responsive as long as there exists a Net N of A × B such that the intersection of N and the universe of discourse (X × Y) is empty, i.e

$$N \cap \underline{\mathbf{X}} = \Phi, \ (\underline{\mathbf{X}} = X \times Y) \tag{20}$$

Proof: Assume that N is the Net of A × B, which satisfies formula (20), i.e

$$\underline{\mathbf{X}} = \underline{\mathbf{X}} - N \tag{21}$$

from formula (21), we derive that

$$\bigvee_{i,j} J_{ij} = \underline{\mathbf{X}} = (X \times Y) \tag{22}$$

so for any definite $(x_o, y_o) \in (X \times Y)$, there exist s, t, such that $(x_o, y_o) \in J_{st}$.

From formula (16), for any $s,i \in I$, $t,j \in J$, if $(s, t) \neq (i, j)$, then

$$A_s(x_o) \wedge B_t(y_o) > A_i(x_o) \wedge B_j(y_o) \tag{23}$$

According to formula (10), the response of the fuzzy controller is as follows

$$C^*(u) = \bigvee_{i,j} (A_i(x_o)) \wedge B_j(y_o) \wedge C_{\varphi(i,\, j)}(u) \tag{24}$$

By formula (23), it is obvious that

$$M(C^*) = M(f) \tag{25}$$

Where

$$M(C^*) = \{u \mid C^*(u) = \mathrm{hgt}C^*(u)\} \tag{26}$$

$$M(f) = \{u \mid f(u) = \mathrm{hgt}\, f(u)\} \tag{27}$$

$$f(u) = (A_s(x_o)) \wedge Bt(y_o) \wedge C_{\varphi(s,\, t)}(u) \tag{28}$$

As it is known that $C_{\varphi(i,\, j)} \in \{C_k\} \neq \Phi$ is a distributed fuzzy set whose kernel is

$$\mathrm{Ker}(C_k) = \{u \mid C_k(u) = 1\} \neq \Phi \tag{29}$$

obviously, there exists an interval $L \subseteq (-\infty, +\infty)$ such that

$$
\begin{aligned}
L &= \{u \mid C_{\varphi(s,\,t)}(u) \geq (A_S(x_o)) \wedge B_t(y_o))\} \\
&= \{u \mid f(u) = (A_S(x_o)) \wedge Bt(y_o))\} \\
&= \{u \mid C^*(u) = \mathrm{hgt}C^*(u)\}
\end{aligned}
\tag{30}
$$

therefore the fuzzy controller is responsive and $u_o = M(L)$.

3. The simplification of fuzzy controller algorithm

In the right-hand side of formula (10), there are I × J terms of union operations. The ordinary algorithm does this calculations term by term and is very time consuming. We know from Theorem 1 that when a fuzzy controller is responsive and the defuzzification mean of maximum is used, we only need to calculate the interval L, then the mid-point of L will be the desired output of the fuzzy controller. Thus for all observed value (x_o, y_o), we only have to calculate f(u), only one of the terms in the formula (10). This will simplify the computation algorithm to a great extent.

Let

$$A_\Sigma = \bigvee_{i \in I} A_i \tag{31}$$

$$A_\Sigma(x) = \bigvee_{i \in I} A_i(x) \tag{32}$$

$$B_\Sigma = \bigvee_{j \in J} B_j \tag{33}$$

$$B_\Sigma(y) = \bigvee_{j \in J} B_j(y) \tag{34}$$

where $x \in X\text{-}\{r'_i\}$, $y \in Y\text{-}\{r''_j\}$. The membership functions of $A_\Sigma(x)$ and $B_\Sigma(y)$ are shown in Figure 3-a.

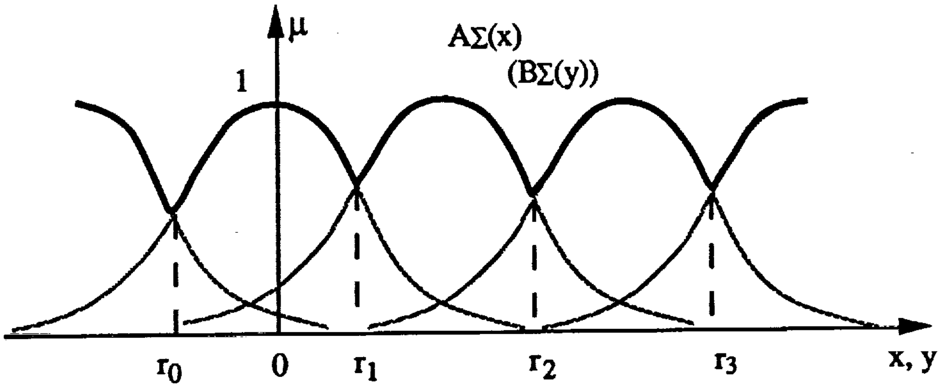

Figure 3-a The membership functions of $A_\Sigma(x)$ and $B_\Sigma(y)$

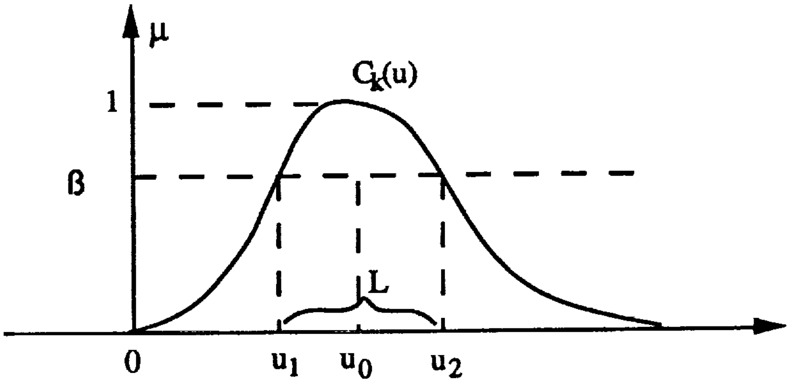

Figure 3-b The function of $\rho_k(\beta)$

Clearly, when $(x, y) \in J_{st}$, i.e $x \in \{r'_{s-1}, r'_s\}$, $y \in \{r''_{t-1}, r''_t\}$

$$A_\Sigma(x) = A_s(x) > A_i(x), \tag{35}$$

$$B_\Sigma(y) = B_t(y) > B_j(y), \tag{36}$$

$$A_\Sigma(x) \wedge B_\Sigma(y) = A_s(x) \wedge B_t(y) > A_i(x) \wedge B_j(y) \tag{37}$$

$$\forall\, s,i \in I,\ t,j \in J,\ (s, t) \neq (i, j)$$

Define the separating functions $\varphi_1(x)$, $\varphi_2(y)$ respectively as follows,

$$\varphi_1(x) = i, \quad x \in \{r'_{i-1}, r'_i\}, \tag{38}$$

$$\varphi_2(y) = j, \quad y \in \{r''_{j-1}, r''_j\}. \tag{39}$$

For $C \in \{C_k\}$, $k \in K$, we define the following function

$$\rho_k(\beta) = M(L) = M(C_{k\beta}), \;\; \beta \in [0, 1] \tag{40}$$

where $L \subseteq (-\infty, +\infty)$,

$$L \cap U = \{u \mid C_k \geq \beta\} \tag{41}$$

where $C_{k\beta}$ is the β- cuted set of C_k and $M(\cdot)$ represents the mid-point of $(\cdot)$ as shown in Figure 3-b. Since $C_k (k \in K)$ is normally distributed set, $\rho_k(\beta)$ is a continuous single-valued function of β, $\forall\, \beta \in [0, 1]$

So far as the functions $A_\Sigma(x)$, $B_\Sigma(y)$, $\varphi_1(x)$, $\varphi_2(y)$ and $\rho_k(\beta)$ are defined, we can derive the following simplified algorithm for the responsive fuzzy controllers:

1) Given the inputs (x_o, y_o) of the fuzzy controller, calculate

$$a = A_\Sigma(x_o), \;\; b = B_\Sigma(y_o), \tag{42}$$

$$s = \varphi_1(x_o), \;\; t = \varphi_2(y_o), \tag{43}$$

$$\beta = a \wedge b = \min(a,b). \tag{44}$$

2) Calculate

$$k = \varphi(s, t), \;\; k \in K \tag{45}$$

where the φ is determined by the given control rules.

3) Finally, the output of the fuzzy controller can be obtained from

$$u_o = \rho_k(\beta) \tag{46}$$

Obviously,

$$\begin{aligned} L \cap U &= \{u \mid C_k \geq \beta\} \\ &= \{u \mid C_{\varphi(s,\, t)} \geq (A_s(x_o)) \wedge B_t(y_o))\} \end{aligned} \tag{47}$$

$$u_o = \rho_k(\beta) = M(L)$$

It can be seen that the result is exactly the same as that in formula (30).

The conventional fuzzy controller algorithm is very time consuming and needs large memory space so that it is hardly possible to implement the fuzzy composition inference on line in a control system. In many applications, fuzzy controllers used look up tables instead of real time inference. Not only it is impossible to tune the fuzzy control rules on line, it takes a great amount of computation time to calculate the fuzzy controller look-up table. The simplified algorithm proposed above reduces the computation greatly and its calculating time is nearly the same as that taken by the conventional PID control algorithm. This makes it possible to do real time fuzzy inference in the controller, allowing the tuning of control rules on line. If the algorithm is used to calculate the fuzzy control look-up table, it takes less than one minute. Since we only need to store 5 functions, namely $A_\Sigma(x)$, $B_\Sigma(y)$, $\varphi_1(x)$, $\varphi_2(y)$ and $\rho_k(\beta)$ instead of all the Ai(x), Bj(y), and Ck(u), a total of I+J+K functions.

4. Hardware Implementation

Fuzzy controller has been highly developed and come to a new stage of hardware implementation. Many fuzzy controllers(or so called fuzzy inference machines) in hardware are available in the market[4][5]. The conventional fuzzy inference algorithm on which most fuzzy controllers are based on is too complicated. Further, its hardware implementation is very expensive and of a large volume, and the inference speed is limited. Reducing its cost and volume and improving its inference speed are very important to this technology.

As can be seen from the last section that with the proposed algorithm, the calculation is much simpler as there is no computation of fuzzy sets and most of the calculations involve only function operations and comparative operations. Therefore, this fuzzy control algorithm is very easy to implement in hardware. The main issue in a hardware design is to construct some function generators generating $A_\Sigma(x)$, $B_\Sigma(y)$, $\varphi_1(x)$, $\varphi_2(y)$ and $\rho_k(\beta)$, while the complicated fuzzy set operation which is difficult to turn into hardware counterparts is avoided.

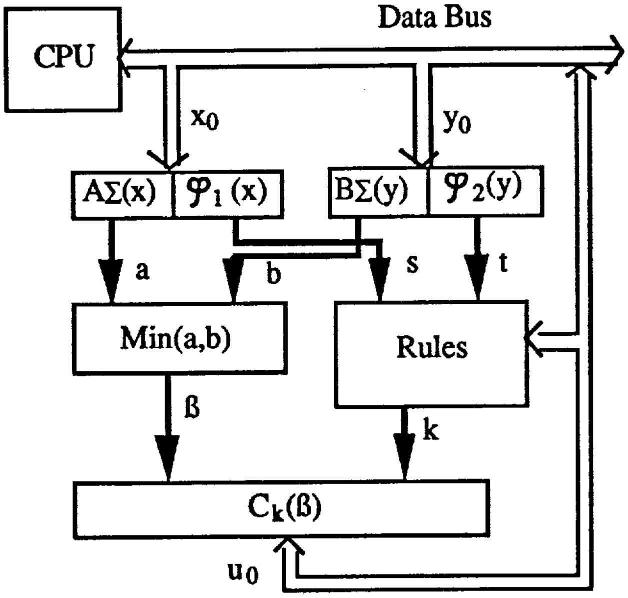

Figure 4 Block diagram of the fuzzy controller board

We have designed a fuzzy controller board for Personal Computers(PC) based on the above algorithm. The principle of the fuzzy controller board is illustrated in Figure 4. The board is composed of some function generators to generate $A_\Sigma(x)$, $B_\Sigma(y)$, $\varphi_1(x)$, $\varphi_2(y)$ and $\rho_k(\beta)$, a comparator to do the operation of Min(a, b) and a rule base to store the control rules. Each part is constructed with digital IC. The detailed design of the hardware will be presented in depth in our future papers.

The controller board is connected to the CPU through the data bus of the PC. The generators of $A_\Sigma(x)$, $B_\Sigma(y)$, $\varphi_1(x)$, $\varphi_2(y)$ and $\rho_k(\beta)$ and the control rules can be programmed conveniently. Using this board with its software environment on a personal computer, it is very flexible to construct a fuzzy control system for an industrial process in which large number of data needed to be processed. This is the reason why we design a fuzzy controller board instead of an independent fuzzy controller machine which is unable to process data and information.

Due to its fuzzy inference function and ability of data processing, the fuzzy control system can be applied not only to the control system but also to many other areas such as expert systems, pattern recognition and decision making where the fuzzy inference method may be employed.

Reference

[1] G. Q. Chen, Analysis into the fuzzy control algorithm, Information and Control, No.5(1980)

[2] M. Mizumoto, Fuzzy reasoning with "if...then else", Applied sys. and cybern., 1, 2927-2932(1980)

[3] P. Z. Wang and S. P. Lou, The responsibility of a fuzzy controller, IFAC 8th Triennial World Cong., Kyoto, Japan, August 1981.

[4] T. Yamakawa, Fuzzy Controller Hardware System, Preprints of 2nd IFSA Congress, Tokyo, July 1987.

[5] T. Yamakawa, A Simple Fuzzy Computer Hardware System Employing Min & Max operation--A Challenge to 6th Generation Computer, Preprints of 2nd IFSA Congress, Tokyo, July 1987.